

AMAIL<sup>tm</sup>  
I  
G  
A

October '86

## SUPPORT SERVICES

Amiga developers can order the following documents from Commodore:

A-1000 Hardware Schematics & Expansion Specifications

Price - \$20

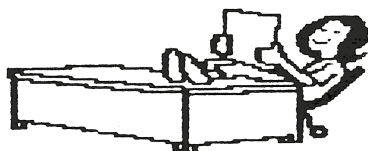
Standard for Interchange Format Files (EA IFF) Documentation & Disk

Price - \$20

To order, send a check payable to  
COMMODORE BUSINESS MACHINES to:

CBM  
ATTN: Kim Montgomery  
1200 Wilson Drive  
West Chester, PA 19380

Be sure to include your name and address with a note requesting the particular item.



## Did You Know ? ? ?

ASDG, Inc. recently released the following hardware products for the Amiga:

1. A RAM expansion board that is user-upgradable, has no wait states, and conforms to auto-config. Introductory prices are:  
0.5 MB - \$395.00  
1.0 MB - \$595.00  
2.0 MB - \$795.00

Members of any Amiga user's group may take an additional 5% off.

2. Mini-Rack B (for Budget) which is a two-slot Zorro expansion backplane and power supply. The price is \$225.00 with the purchase of a memory board.

For more information on these products, call Perry Kivolowitz at ASDG, Inc. (201-540-9670)

You've got a friend in  
**A 1000**  
Pennsylvania

If you have a question regarding Amiga development which could be addressed in our newsletter, send it to:

Becky Cotton  
Software Department  
Commodore Business Machines  
1200 Wilson Drive  
West Chester, PA 19380

Amiga developers who would like to add Workbench, Header or Include files in their products must secure a licensing agreement from Commodore. A \$100/per year fee will be charged to process the agreement. Write to Kim Montgomery at Commodore's Software Department to obtain the license.



This month's newsletter includes your invitation to the Second Annual Amiga Developer's Conference. Several developers have already registered, so don't delay in returning your registration form. The following are among some of the many companies that have reserved booth space to demonstrate new Amiga products:

Argonaut	Microbotics
ASDG, Inc.	New Horizons
Electronic Arts	Proprietary Products
Maxisoft	Tid Bit Software
Metadigm	Woodsoft

## Version 1.2 Upgrade

Kickstart and Workbench version 1.2 will be distributed to dealers approximately one month after our developers receive the approved final release version. Release version 1.2 is currently in testing but is expected to pass within the next few weeks. Developers who have received the pre-release version of 1.2 will also receive the final version.

## A Reminder About... Library Version Checking

If your software product uses new 1.2 functions and features, it just might get a visit from the Guru on a machine booted with 1.1.

This could make a great product look bad.

Keep the Guru away by specifying the required library versions in your OpenLibrary calls. Use the number of the earliest version that contains the functions and features you require. The OpenLibrary ( ) will fail (return 0) if the machine is booted with an older version of the operating system. You should test for such failures and abort cleanly, informing the user that the software requires a particular version (1.1, 1.2, etc.).

The version numbers for use in OpenLibrary ( ) are as follows:

0	= any version
30	= 1.0 or higher
31	= 1.1 or higher
32	= Pal release or higher
33	= 1.2 or higher

Carolyn Scheppner  
Commodore-Amiga

# AMIGA PROMOTION MEDIA SCHEDULE

	OCTOBER	NOVEMBER	DECEMBER
	29 6 13 20 27	3 10 17 24	1 8 15 22
<b>NATIONAL NEWSPAPER</b> (Commodore Sponsored)			
<b>USA Today</b>	■	■	
<b>Regional &amp; Local</b>	■	■	
<b>LOCAL NEWSPAPER</b> (Dealer Co-Op)	■	■	
<b>SPOT RADIO</b> (Dealer Co-op)	■	■	
<b>GRAPHICS MARKET</b> (Trade Magazines)			
<b>Communication Arts</b>			■
<b>Art Product News</b>		■	■
<b>Art Direction</b>			■
<b>Advertising Age</b>		■	■

```

/*****
*          ----- auto_req.c -----
*   Here's a routine to automagically construct AutoRequests.
*   BuildAutoRequest, workbench routine by Neil Katin
*
*   Modified to stand alone by Andy Finkel
*   And Jim Mackraz made the suggestion that it be broken out
*
*   Modified 07/25/86 by Lisa Siracusa to work with Lattice C
*****/

# include "exec/types.h"
# include "exec/nodes.h"
# include "exec/lists.h"
# include "exec/memory.h"
# include "exec/alerts.h"
# include "clib/macros.h"
# include "graphics/rastport.h"
# include "libraries/dos.h"
# include "libraries/dosextens.h"
# include "intuition/intuition.h"

/*****
*   BuildAutoRequest
*       takes window pointer,
*       array of strings to be displayed,
*       text for positive gadget,
*       text for negative gadget,
*       positive flags (see AutoRequest, in Intuition manual)
*       negative flags (see AutoRequest, in Intuition manual)
*****/

BuildAutoRequest(w, textArray, posText, negText, posflags, negflags)
struct Window *w;
char **textArray;
char *posText;
char *negText;
ULONG posflags;
ULONG negflags;
{
    struct IntuiText *itArray,
                    *ita,
                    *itPos,
                    *itNeg;

    SHORT  fontSize;
    char **sa;
    int    height,
           width;
    int    sizeArray;
    int    result;

    /* make sure there is at least one string */
    if(! *textArray)
        return(0);

    /* compute height and width */

```

```

width = textLength(w,posText) + textLength(w,negText) + 70;
fontSize=(w->WScreen->Font->ta_YSize);
height = fontSize * 2 + 22;

for(sa = textArray; *sa; sa++)
{
    height += fontSize + 2;
    width = MAX(width,textLength(w,*sa) + 34);
}

sizeArray = ((sa - textArray) + 2) * sizeof(struct IntuiText);

/* allocate intuitext structures */
itArray = (struct IntuiText *) AllocMem(sizeArray, MEMF_CLEAR);
if(! itArray)
    return(0);

/* now initialize the intuitext structures */
ita = itArray;
sa = textArray;

ita->TopEdge = 4;
for(ita = itArray, sa = textArray; *sa; sa++, ita++)
    initIntuiText(fontSize, ita, *sa, &ita[1]);

/* keep the next one from pointing off into space */
ita[-1].NextText = NULL;

itPos = ita++;
initIntuiText(fontSize, itPos, posText, 0);

itNeg = ita;
initIntuiText(fontSize, itNeg, negText, 0);

itPos->TopEdge = itNeg->TopEdge = AUTOTOPEDGE;
itPos->LeftEdge = itNeg->LeftEdge = AUTOLEFTEDGE;

result = AutoRequest(w, itArray, itPos, itNeg,
                    posflags, negflags, width + 16, height);

FreeMem(itArray, sizeArray);
return(result);
}

```

```

initIntuiText(size, it, text, itNext)
SHORT size;
struct IntuiText *it;
char *text;
struct IntuiText *itNext;
{
    it->FrontPen = 0;
    it->BackPen = 1;
    it->DrawMode = JAM1;
    it->IText = text;
}

```

```
if(itNext)
{
    it->NextText = itNext;
    itNext->TopEdge = it->TopEdge + size+2;
    it->LeftEdge = 12;
}

}

textLength(w,string)
struct Window *w;
char *string;
{
return(TextLength(w->RPort,string,stringlen(string)));
}

stringlen(s)
register char *s;
{
register i = 0;

while(*s++)
    i++;
return(i);
}
```



```

/*****
 * This program is the test program for auto_req.c. It
 * is compiled seperately, however it must be linked
 * with the auto_req.o file.
 *****/

# include "exec/types.h"
# include "exec/execbase.h"
# include "exec/exec.h"
# include "graphics/gfxbase.h"
# include "graphics/rastport.h"
# include "graphics/layers.h"
# include "intuition/intuition.h"

# define BLUE 0
# define WHITE 1
# define BLACK 2
# define RED 3

extern struct Window *OpenWindow();

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct Window *window;

struct NewWindow nw =
{
    100, 10, /* starting position (left,top) */
    300, 100, /* width, height */
    BLUE, WHITE, /* detailpen, blockpen */
    /* idcmp flags */
    NULL,
    /* window gadget flags */
    WINDOWDEPTH|WINDOWSIZING|WINDOWDRAG|ACTIVATE,
    NULL, /* pointer to 1st user gadget */
    NULL, /* pointer to user check */
    "BuildSysRequest Test", /* title */
    NULL, /* pointer to window screen */
    NULL, /* pointer to super bitmap */
    100, 45, /* min width, height */
    640, -1, /* max width, height */
    WBENCHSCREEN
};

char *discardstrings[] =
{
    "Warning: you",
    "cannot get back",
    "what you discard",
    0
};

main()
{
    int status;

```

```

if((IntuitionBase = (struct IntuitionBase *) OpenLibrary("intuition.library",0))
    Error();
if((GfxBase = (struct GfxBase *) OpenLibrary("graphics.library", 0)) == NULL)
    Error();

if((window=OpenWindow(&nw)) == NULL)
    Error();
/* kprintf("\nwindow is open\n"); */ /* to use link with debug.lib */

status= BuildAutoRequest( window, discardstrings, "ok to discard", "forget it!",
/* kprintf("status= %ld\n",status); */ /* to use link with debug.lib */

Cleanup();
exit(0);
}

Cleanup()
{
if(window)CloseWindow(window);
}

Error()
{
Cleanup();
exit(1);
}

```



# 2nd Annual Amiga Developers Conference

*November 5 – 7, 1986*

*Monterey, California*

*Double Tree Inn*

Commodore-Amiga cordially invites you to attend the 2nd Annual Amiga Developers Conference. The success of last year's conference has inspired us to return to beautiful Monterey. During these three days you will become familiar with the latest in Amiga Technology, meet the people responsible, and make new acquaintances in the Amiga Developers Community. Also, developers will have an opportunity to meet with Amiga Sales Representatives. We will be providing, on a first come first serve basis and at no additional charge, booth space at the conference FAIRE where developers can introduce new products or demonstrate existing ones. The press will be on hand for this event! If you plan on making a press announcement during this event, please fill out and return the attached Press Announcement Form. It is important that each organization get their fair share of the available time for press announcements.

The fee is \$300.00 per person for the full conference which includes:

- Lectures
- Workshops
- Cocktail Party, November 4th
- All Conference Materials
- Admission to and participation (if requested) in the FAIRE
- Awards Banquet, Friday, November 7th

Each attendee will be responsible for his/her own room, transportation and meals except those listed above.

Please fill out and return the registration form on or before **October 17, 1986**. Payment must accompany registration.

If you wish, you may attend Friday night's **USERS GROUP AWARDS BANQUET** only. The cost is \$40.00 which includes a buffet dinner and an open bar. The bar will open at 5:30 p.m., dinner served at 6:30 and awards will be given for products released for the Amiga as selected by users of the Amiga. Balloting for these awards is being handled independently from Commodore-Amiga.

**We will be accepting reservations on a first come first served basis, so please register early.**

Attendees should arrange for their own transportation and hotel reservation. If you would like to stay at the Double Tree a block of rooms has been reserved for our Conference in the name of Commodore-Amiga. All you have to do is to call and ask for one of the rooms being held in our name. The phone number of the Double Tree is (408) 649-4511.

*Registration Form*  
*2nd Annual Amiga Developers Conference*  
*Monterey, California*

Full Conference                      \$300.00 per person

Awards Banquet Only              \$ 40.00 per person

Please complete this form and return to Commodore-Amiga prior to October 17, 1986:

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Phone Number \_\_\_\_\_

I am including \$\_\_\_\_\_ for (check one) \_\_\_\_\_ full conference \_\_\_\_\_ Banquet Only

Please indicate if booth space is required. Information regarding FAIRE participation will be sent to you under separate cover.

YES, I would like a booth. \_\_\_\_\_

Mail to:                      CARYN MICAL  
                                 Commodore-Amiga  
                                 983 University Avenue  
                                 Los Gatos, California 95030

Confirmed registrations cancelled less than one week prior to the conference are subject to a \$50 service charge.

Registrants with confirmed applications who fail to attend are liable for the entire conference fee.

Substitution of attendees can be made up to one week prior to the conference.

Please submit one registration form per attendee. Copy form if necessary.

**this document was  
generously  
contributed by**

**randell jesup**